Self-Organizing Networks in LTE: a Q-learning approach to ABS optimization

Andrea Maracani, Marco Rossanese, Davide Talon
Department of Information Engineering, University of Padova – Via Gradenigo, 6/b, 35131 Padova, Italy
Email: {andrea.maracani,marco.rossanese,davide.talon}@studenti.unipd.it

***Abstract*—In this paper we explain-propose a possible machine-learning approach for the enhanced inter-cell interference coordination (eICIC) in a heterogeneous network (HetNet), where macro and micro cells optimize their downlink transmission in a self-adaptive manner. The idea is to exploit the Q-learning algorithm to guarantee a fair resource sharing between micro and macro cells and so to make the system works in the best performance possible. Indeed this specific reinforcement learning technique is used to find out the optimal policy that leads to the right ABS pattern setting, designed specifically for each scenario. Hence evaluating only a few parameters and changing dynamically the ABS pattern we aim to cope the possible alterations on the environment.**

***Index Terms*—SON, Self Optimizing Networks, LTE, eICIC, ABS, reinforcement learning, Q-learning, MONSTeR**

I. Introduction

The great spreading of mobile devices all over the world represents the fastest adoption of any technology that our society has ever experienced, faster than the Internet and the earlier generations of mobile communications. Now tablets, android devices, iPhones, application stores, social media and the data exchanges between end-users and clouds are all growing at exponential speed. Providing the necessary bandwidth and capacity to the people so they can keep the pace of this growth is a fascinating challenge; then in order to achieve that, a more network densification is required together a full exploitation of the simultaneous presence of micro (coverage radius around 10m-300m) and macro (coverage radius up to 20km) cells, the so called heterogeneous networks (HetNet).

add statistics and forecast of mobile traffic The only way these challenges can be cost-effective, efficient and human-handleable is through the use of more automated and autonomous systems, such as Self-Organizing Networks (SON): the goal is to minimize the human intervention in the planning, deployment, optimization and maintenance activities of these new networks.

A such SON conceptually must own, as explained by [5], the following capabilities:

- Self-Planning: process of identifying the parameter settings of new network elements (like radio parameters of a new eNodeB or a table of neighbor nodes);
- Self-Deployment: preparation, installation, authentication and delivery of a status report of a new network node in order to get a "plug and play" approach for each new device;
- Self-Healing: execution of the routine actions that keep the network operational and/or prevent problems (this includes the necessary software and hardware upgrades);
- Self-Optimization is defined as the utilization of measurements and performance indicators collected by the User Equipments (UEs) and the base stations in order to auto-tune the network settings.

Our work is focused on the last point of the list, more precisely on the improvement of the signal quality and consequently the throughput in a LTE system, trying to minimize the interference between micro and macro cell with an adaptive coordination in downlink transmissions of the antennas (mainly controlling the transmit power patterns).

The problem we're facing consists in discovering an optimal trade-off of the radio resources to be assigned between high-power nodes (macro) and low-power nodes (micro). The latter has an higher capacity and consequently an user connected to it will benefit of a better performance, however due to fact that it shares the same frequency band with the macros, low-power nodes are severely affected by the interference from the high-power nodes. In other words, downlink micro transmissions to its UEs could be sorely degraded by high power macro transmission; besides UEs that are close to a micro could end up associating to a macro due to the higher power strength received from the stronger node.

So the issue is that, accounting the above scenarios, the micro could be left underutilized and this would turn out to be a bad exploitation of the resources deployed appositely to enhance the connectivity in some areas, with a following shortage of the user performance in comparison with achievable potential capacity.

Addressing a fair resource sharing, a solution could be stop macros from all the transmissions (some exception signals, like beacons, are always active), for a certain amount of time: this approach is adopted by LTE 3GPP standard and is known as enhanced Inter Cell Interference Coordination (eICIC). The period in which macros are "silent" is called Almost Blank Subframe (ABS), period over which micros can transmit with reduced interference.

Time is slotted in frames of 10 ms and each one of these is split into 10 subframes, so a macro decides to mute itself in a certain subframe following the ABS mask, that's the pattern of 0's and 1's where 1 indicates the macro's inactivity.

The goal is to discover an adaptive ABS mask that maximizes the UEs' performance and it's found as the result of an optimization problem. Our idea is to exploit a reinforcement learning (RL) approach, the so called Q-learning technique in order to reach the fairness between the shared resources and set the best ABS mask in a self-tuning manner for each specific scenario.

Thanks to the LTE environment simulator MONSTeR (MObile Networks SimulaToR) [1], a framework built around the LTE system toolbox available in Matlab, we simulate a great amount of data traffic with the purpose of training the Q-learning: after this period the method involved will yield the best policy for setting the considered mask.

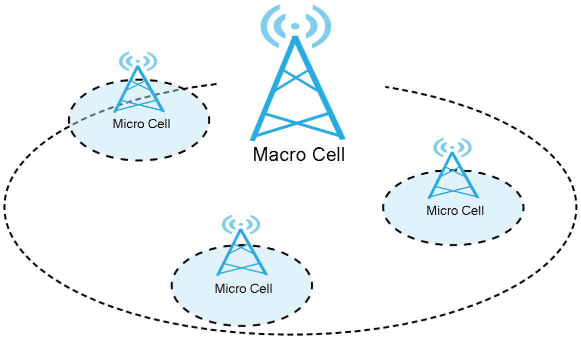

Figure 1: Example of a typical HetNet with a macro and some micros used for throughput enhancement in high density utilization area; a micro is placed on the edge of the coverage area for improving the border throughput

TO DO (REMEMBER TO ERASE THIS)

- **Results: summary of the main findings**

II. Related Work

In this section we will overview the main proposals in the literature.

First of all we must take in account the solution exposed in [2] to protect the downlink micro transmissions. Alongside the concept of ABS period they introduced the so called Flexible User Association and Cell selection bias (CSB). As we already know, whenever a UE needs to select a suitable cell for association it chooses the one with the strongest signal and therefore the utilization of the micro mostly of time is neglected.

LTE has introduced the bias value α_i, that is broadcasted to all the UEs. Hence the UE will choose the right cell maximizing the sum between α_i and the received signal power P_i over each i-th cell in the system; obviously higher bias values are assigned to the micros.

The authors of [2] proposed two algorithms, one for the optimal ABS mask (OPT-ABS) and one for defining the right value for the CBS. The former is split into two parts: the first to search the number of optimal subframes in which the macros are mute and second how to define the optimal ABS pattern. This algorithm turned up to be a NP-hard computational problem, however they proofed that with a cleaver breakdown in some defined steps it's implementable and the complexity is scalar with number of cells. Instead the latter algorithm compute the biases so that the "association error" (as compared to optimal association) is minimized. This technique shows some shortcomings as the fact that is hardly applicable to a non-static scenario because the beforehand knowledge of the locations is required in order to evaluate which is the best cell in the area, information not always available.

Another noteworthy paper is surely [6], here the authors tried to put into practice a similar reinforcement learning approach. The state is made of two parameters, the Signal to Intererence Noise Ratio (SINR) for the player in relation with the macro and the micro, nevertheless these are not values that the eNodeB takes normally in account and we are aiming to use only the minimum number of information that are already exchanged, avoiding the loss of time due to computation for calculations (all the info exchanged between eNodeB and UE taken from the 3GPP release [3]).

III. System Model

In this section we will explain all the operative assumptions that we took in account for our proposal and the steps followed in our work.

A. Problem modelization

As stated in [7], the main book about RL: the reinforcement learning problem is meant to be a straightforward framing of the problem of learning from interaction to achieve a goal. The learner, who is the one that makes the decisions is called the agent and in this scenario will be the macro. The thing it interacts with, comprising everything outside the agent, is called the environment, part played here by the simulator MONSTeR. These interact continually, the agent selecting actions and the environment responding to those actions and presenting new situations to the agent. The environment also yields rewards, numerical values that evaluate the goodness of the last action taken and that the agent tries to maximize over time. At each time step t, the agent receives a representation of the environment's state S_t, here defined by three parameters:

- the ratio between the number of users in the macro and in the micro, quantized in four values which indicates if there're much more users in the macro, a little bit more in the macro, a little bit more in the micro or much more in the micro;
- the ratio between the throughput of macro and mean value of all micros, also here quantized but in five values;
- the number of ABS quantized in six values where each of them stays for the amount of 1's, always a multiple of 2, and those 1's are distributed in a certain pattern in the mask;

so we assume that the total number of possible states in our environment is exactly $4 \cdot 5 \cdot 6 = 120$. Depending on which state the agent is, it selects an action A_t from the set of all actions $A = \{-2, 0, 2\}$ that will modify the ABS mask by this amount. The function reward selected for our system is

$$r(n^m, n^M, S^m, S^M) = -\left(\frac{n^m \cdot S^M}{n^M \cdot S^m} - 1\right)^2$$

where m stands for micro and M for macros and it shows the discrepancy between the actual system condition and the

equilibrium we're aiming to.
At each time step, the agent maps from states to probabilities of taking each possible action. This mapping is called policy and is denoted as π_t. Reinforcement learning specifies how the agent changes its policy in relation of its experience. The agent's goal so is to maximize the total amount of reward it gains on the long term.

B. *Q-learning technique*

Q-learning algorithm was proposed by Watkins in [8] and consists in maintaining a table of estimated expected long-run reward, called indeed Q-values and written as $Q(S, A)$, for each state-action couple. This technique is classified between the off-policy methods, in other words it evaluates or improves a policy different from that used to generate the data, as explained by [7]. So the idea to exploit two policies, one that is learned about and that becomes the optimal policy (target policy), and an exploratory one used to estimate the Q-values (exploration policy). Despite that the most used policies to implement the latter are the ϵ-greedy or the softmax, we decided to use a totally random selection policy.
After Q-values have been enough discovered, the Q-learning function points out how to update the Q-value at each step and it's described as follows:

$$Q(S_{t+1}, A_{t+1}) = Q(S_t, A_t) + \alpha[R(S_t, A_t) - Q(S_t, A_t)]$$

where R is the maximized expected reward and α indicates with which rates we take actions towards the best Q-values achievable.

IV. Simulation Settings

In the present work in order to evaluate the proposed algorithm we run out an extensive Matlab simulation using MONSTeR. We generated an urban scenario of 9000 squared meters with 7 horizontal and 6 vertical streets and among them we located buildings with height uniformly distributed from 20 to 50 meters. Then only a macro eNodeB is placed at the center of the city and 3 micros are equally distributed with distance from the macro of **40 m**; the former has height 35 m and 50 Resource Blocks (RB) and the latter have height 25 m and 25 RBs. The mobility of the users is generated randomly, 15 users move with velocity **70 m/s** along sidewalks and at each cross have the same probability to turn left/right or to go straight ahead.
To simulate the channel and the wireless communication we considered the Winner II channel model with **dense urban** propagation scenario implemented inside Matlab Communications System Toolbox. To assess the worth of our algorithm we considered as main metrics the Bit Error Rate (BER) and the average throughput of involved users therefore we compared the proposed optimization with other 2 ABS choice policy in 3 different traffic scenarios:

- **Static ABS**: the number of Almost Blank Subframes remains constant to 4 over the entire simulation and cells use the same ABS mask [0,1,0,1,0,1,0,1,0,0];
- **Random ABS**: at each frame, macro cell randomly selects the ABS mask choosing over three possible action: increase, decrease or keep constant the number of almost blank subframes. Each action has the same probability, however if we are dealing with 0 ABS the only possible actions are to keep constant or increase with propability respectively 1/3, 2/3. The case of 10 ABS is handled analogously but with descreasing action in lieu of increasing.

Table **??** sums up the ABS policy parameters.
The packet generation event from an user is modelled as a Possion process of rate λ which is taken from a Gaussian distribution $\lambda \sim \mathcal{N}(10, 5)$. So whenever a generation event occurs a packet file is created to be sent to the eNodeB with a certain probability inversely proportional to the size $\{$**to be defined** $\}$bit, i.e. the smallest packet will benefit of an higher likelihood due to the fact that we're simulating a realistic human behavior during a web surfing period.

Table I: ABS policy algorithm parameters

ABS policy	Parameter	Value
Random policy	$Prob[a_{incr}]$	5
	$Prob[a_{cost}]$	3
	$Prob[a_{decr}]$	3
Static	N_{ABS}	30 s
Q-learning	maximum buffer size	30 s
	maximum buffer size	30 s
	γ	2
	α	2

A. *Training Phase*

The learning algorithm was trained over **N** different simulations composed of 125 frames of 10 ns for a total of **N*125** frames and simulated time **N*125*10ns** seconds. Using the off-policy propery of the Q-learning algorithm we ensured that all possible states were visited several times such that the value functions for the pair state-action could be well approximate.

B. *Test Phase*

After all these necessary premises, the first step is to run a big amount of simulations with the purpose of explore all the couples state-action, the Q-values, following the random policy and then the Q-learning will be applied to find out which will be the optimal policy to take. So at this point a several number of simulations with the standard ABS policy and with optimal one will be run to compare the results in terms of throughput (and delay?).

V. Results

The Results section contains a selection of the most relevant results with the explanation of their meaning. Please, not that you do NOT have to describe the shape of the curves that can be seen in the figures, but the reasons WHY such curves have that shape!

VI. Conclusions

Conclusions are a superbrief summary of what has been done and highlighting of the "take home message"

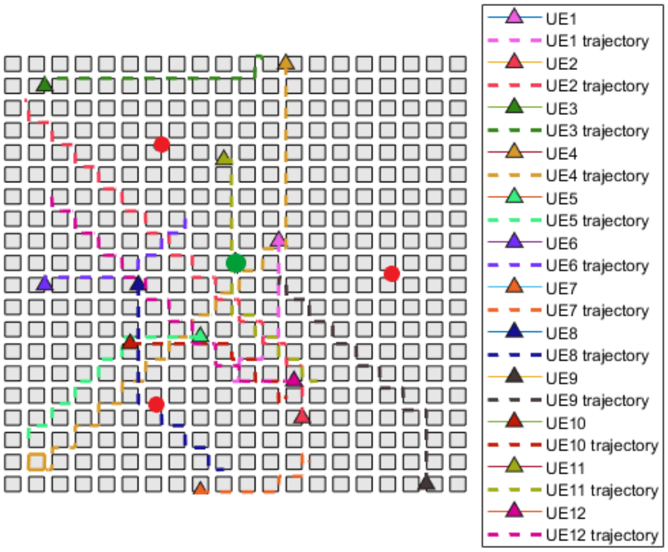

Figure 2: Example of UEs in the urban simulated environment and highlighted trajectories; red dots represent the micros and the green the macro.

References

[1] Monster: a framework built around the matlab lte system toolbox. https://github.com/Sonohi/monster, 2017.

[2] Supratim Deb, Pantelis Monogioudis, Jerzy Miernik, and James P Seymour. Algorithms for enhanced inter-cell interference coordination (eicic) in lte hetnets. *IEEE/ACM transactions on networking*, 22(1):137–150, 2014.

[3] TS ETSI. 136 211. *Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 general aspects and principles (3GPP TS 36.420 version 14.0.1 Release 14)*, 2017.

[4] David Lopez-Perez, Ismail Guvenc, Guillaume De la Roche, Marios Kountouris, Tony QS Quek, and Jie Zhang. Enhanced intercell interference coordination challenges in heterogeneous networks. *IEEE Wireless Communications*, 18(3), 2011.

[5] Juan Ramiro and Khalid Hamied. *Self-organizing networks (SON): Self-planning, self-optimization and self-healing for GSM, UMTS and LTE*. John Wiley & Sons, 2011.

[6] Meryem Simsek, Mehdi Bennis, and Ismail Guvenc. Enhanced intercell interference coordination in hetnets: Single vs. multiflow approach. In *Globecom Workshops (GC Wkshps), 2013 IEEE*, pages 725–729. IEEE, 2013.

[7] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[8] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.